11. Уразаев Р.П. Методы генерации алгоритмов прогнозирования при помощи операций над базовыми алгоритмами [Текст] / Р.П. Уразаев. – М. : Вычислительный центр АН СССР, 1988. – 25 с.
12. Одейчук А.Н. Обобщенный критерий эффективности моделей прогнозирования временных рядов в информационных системах [Текст] / А.Н. Одейчук // Біоніка інтелекту: наук.-техн. журнал. – 2009. – № 1 (70). – С. 113-119.
13. Бронштейн И.Н. Справочник по математике для инженеров и учащихся вузов [Текст] / И.Н. Бронштейн, К.А. Семендяев. – [13–е изд.]. – М. : Наука, Гл. ред. физ.–мат. лит., 1986. – 544 с.
14. Одейчук А.Н. Элементы математического обеспечения интеллектуальной системы прогнозирования в условиях гетероскедастичности [Текст] / А.Н. Одейчук // Радиоэлектроника и молодежь в XXI веке : 13-й международный молодежный форум, 30 марта – 1 апреля 2009 г. : материалы форума. – Харьков : ХНУРЭ. – 2009. – Ч. 2. – С. 107.
15. Одейчук А.Н. Интеллектуальная система выбора метода прогнозирования стохастических рядов в условиях гетероскедастичности [Текст] / А.Н. Одейчук, Б.В. Шамша, Є.Г. Федоров // АСУ и приборы автоматики. – 2007. – Вип. 138. – С. 9–14.
16. Odeychuk Andrey. The expert system of search the forecasting method with using of neural network in volatility conditions of initial data [Текст] / Andrey Odeychuk, Olesya Morozova, Anastasiya Gud // Modern problems of radio engineering, telecommunications and computer science: proceedings of the international conference TCSET 2008, 19-23 February 2008. – Lviv : Publishing House of Lviv Polytechnic. – 2008. – P. 55–58.
17. Ивахненко А.Г. Самоорганизация прогнозирующих моделей [Текст] / А.Г. Ивахненко, Й.А. Мюллер. – К. : Техніка, 1985. – 223 с.

УДК 65.011.56

ВЫБОР ПРОГРАММНЫХ ТЕХНОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СЕРВЕРНОЙ ЧАСТИ ХОСТИНГ-ПРОВАЙДЕРА

А.В. Олейников*
Е.П. Павленко
Кандидат технических наук, доцент*
E-mail: evg-pavl@mail.ru
*Кафедра информационных управляющих систем**
В.А. Айвазов
Старший преподаватель
Кафедра охраны труда**
**Харьковский национальный университет радиоэлектроники
пр. Ленина, 14, г. Харьков, 61166
Контактный тел.: (057) 702-14-51

Досліджені основні переваги та недоліки програмних технологій Spring Framework та Enterprise Java Beans. Розглянута задача обліку користувачів та серверного парку хостинг провайдера та була обрана технологія для її вирішення

Ключові слова: Spring Framework, Enterprise Java Beans, хостинг-провайдер

Исследованы основные преимущества и недостатки программных технологий Spring Framework и Enterprise Java Beans. Рассмотрена задача учета пользователей и серверного парку для хостинг-провайдера и выбрана технология для её решения

Ключевые слова: Spring Framework, Enterprise Java Beans, хостинг-провайдер

Advantages and disadvantages of server's technologies EJB and Spring Framework were analyzed. The problem of servers and customers accounting on "Byte Host" hosting provider was considered and the technology for solving this problem was chosen

Keywords: Spring Framework, Enterprise Java Beans, a hosting provider

1.Введение

Современный бизнес требует все более широкого применения информационных технологий в управлении предприятием. Жизнеспособность и развитие информационных технологий объясняется тем, что современный бизнес крайне чувствителен к ошибкам в управлении. Для принятия любого грамотного управленческого решения в условиях неопределенности и риска необходимо постоянно держать под контролем различные аспекты финансово-хозяйственной деятельности. Поэтому современный подход к управ-

лению предполагает вложение средств в информационные технологии.

Комплексная автоматизированная обработка информации предполагает объединение в единый комплекс технических средств обработки информации с использованием новейшей технологии, методологии и различных процедур по обработке информации. Создание комплексной автоматизированной системы хостинг-провайдера предполагает использование всего комплекса технических средств обработки информации, переход к единой системе обработки всех видов информации.

2. Постановка задачи

Предприятие "ByteHost" занимается предоставлением услуг VPS (Virtual Private Server), Shared, Colocation хостинга.

Предоставление услуг происходит следующим образом:

- клиент через веб-интерфейс производит выбор тарифного плана, производит заказ;
- клиент производит оплату одним из доступных ему способов;
- в случае возникновения проблем или вопросов, связанных с оплатой, клиент обращается в службу поддержки за консультацией;
- клиент получает в пользование сервер или возможность размещения собственного сервера в дата-центре на оговоренный срок;
- в случае возникновения технических проблем при использовании услуг провайдера клиент обращается в службу технической поддержки, где может получить консультацию или оставить заявку на решение его проблемы в случае возникновения неполадок со стороны провайдера.
- после окончания действия договора клиент по желанию может продлить срок аренды сервера или места.

Предприятие состоит из дата-центра, где находится технический отдел и офиса, где находится руководство компании, отдел поддержки пользователей, бухгалтерия и отдел кадров.

На предприятии возникла необходимость разработать программное обеспечение для серверной части информационной системы хостинг-провайдера. Данная система должна поддерживать роли пользователей, тем самым ограничивая пользователям доступ к информации, а также предоставляя требуемый интерфейс в зависимости от роли пользователя. Клиентская часть системы должна быть построена таким образом, чтобы возможно было использовать существующее аппаратное обеспечение организации без его обновления. Система должна обеспечивать работу сотрудников удаленно, при этом предоставлять возможность комфортно работать при скорости соединения не менее 256kbps.

Перечень функций, которые должна выполнять система: создание заявки на закупку нового оборудования или комплектующих; рассмотрение заявки на закупку нового оборудования или комплектующих с последующим принятием решения о удовлетворении заявки; оформление документов о выдаче средств на закупку аппаратного обеспечения; доступ к сообщениям, предназначенным для сотрудников отдела; составление финансового отчета; вывод аналитической информации; создание резервной копии данных; восстановление данных из резервной копии; ведение архива.

Серверная часть ПО должна запускаться и работать на сервере с конфигурацией не ниже Dual Xeon 3.0Ghz, 2Gb RAM, 500Gb Hdd, 1Gbit Lan.

При разработке серверной части программного обеспечения информационной системы хостинг-провайдера возникла проблема выбора платформы для разработки приложения. Поскольку языком разработки был выбран Java, выбор необходимо было сделать между использованием Spring Framework и Enterprise Java Bean (EJB).

3. Сравнительная характеристика программных технологий Spring Framework и Enterprise Java Beans

Spring и EJB имеют много различий. Прежде всего, EJB – это спецификация, а Spring - это имплементация. Эти две технологии используют совершенно разный структурный подход. Spring framework строится на базе инверсии зависимостей контейнера и является многослойной платформой разработки корпоративных приложений. EJB имеет компонентную архитектуру. Обе технологии могут сравниваться, поскольку Spring framework был разработан как альтернатива EJB.

Обе платформы имеют широкую поддержку технологий объектно-реляционной привязки. Spring framework не предоставляет своей имплементации, однако обладает тесной интеграцией с популярными ORM (Object-relational mapping) фреймворками, такими как Hibernate, JDO, iBatis и Java persistence API (JPA). В то же время EJB явно работает только с JPA, однако JPA в свою очередь является спецификацией и множество платформ имеют ее поддержку. И их число входят Hibernate, Kodo, TopLink и другие.

Схема данных разрабатываемого для хостинг-провайдера приложения представляет собой сложный граф объектов. Если при запросе объекта выбирать из базы данные обо всех связях объекта, то ресурсы расходуются крайне нерационально. Для решения этой проблемы используются «ленивые» запросы. То есть данные получаются из БД по мере необходимости. Проблема состоит в том, что для того, чтобы выполнить «ленивый» запрос, сессия БД должна быть все еще открыта после извлечения основного объекта. Подход Spring – открывать и закрывать сессию во время формирования вида (используется шаблон Model-View-Controller). Для этого предусмотрен класс OpenSessionViewFilter. Подход EJB состоит в выделении дополнительных границ видимости для контекста хранимых данных, что позволяет привязать время жизни сессии к времени жизни объекта.

Важным моментом в разработке Интернет-приложения хостинг-провайдера является управление транзакциями. Его важность обусловлена тем, что в высоконагруженных приложениях вероятность того, что данные будут одновременно изменяться несколькими пользователями, достаточно высока. Для обеспечения работы с транзакциями Spring использует принцип аспектно-ориентированного программирования. Описание прокси-классов могут быть выполнены с помощью xml файлов, а так же используя аннотации. Пример использования аннотаций для управления транзакциями проиллюстрируем следующим фрагментом кода:

```
package me.oleynikovav.purchaseservice.manager.impl;

import me.oleynikovav.purchaseservice.dao.AccountDao;
import me.oleynikovav.purchaseservice.domain.Account;
import me.oleynikovav.purchaseservice.manager.AccountManager;
import me.oleynikovav.purchaseservice.manager.ApplicationManager;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Lazy;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

@Service //Объявление класса как компонент. Контейнер управляет экземплярами
данного класса
@Lazy //Указание контейнеру не создать экземпляр класса по мере надобности, а
не в момент инициализации
public class AccountManagerImpl extends GenericDaoManagerImpl<Account, Long>
        implements AccountManager {
    private static Log LOG = LogFactory.getLog(AccountManagerImpl.class);

    @Autowired
    private AccountDao dao; //Использование аннотаций для получения
    конпонентов из контейнера
    @Autowired
    protected ApplicationManager applicationManager;

    @Override
    @Transactional(propagation = Propagation.REQUIRED) //Использование
прокси для управления транзакцией.
    public void changePassword(String userName, String password,
            String appName, String newPassword) {
        applicationManager.getValidApplication(appName);
            Account account = getAuthorizedAccount(userName, password,
            appName);
        account.setPassword(newPassword);
        saveOrUpdate(account);
        if (LOG.isTraceEnabled()) {
                LOG.trace("Password changed for account: " +
                account.toString());
        }
    }
}
```

Управление транзакциями в EJB привязано к менеджеру ресурсов, потому подход к обработке транзакций определяется видом ресурсов, обращение к котором производится.

Поддержка состояний является важным элементов приложения хостинг-провайдера, поскольку взаимодействие с клиентом часто выполняется в несколько шагов. При регистрации в системе заполняется несколько форм, а затем требуется объединить информацию со всех форм, заполненных пользователем. Подходы Spring и EJB в реализации данной функции существенно отличаются друг от друга. EJB предлагает механизм Stateful Session Bean (SFSB), который позволяет сохранять состояние объекта между запросами. Этот подход позволяет оптимизировать производительность сервера путем балансирования данных SFSB между оперативной памятью и жестким диском. Недостатком такого подхода является то, что он не так хорошо подходит для приложений, развернутых на нескольких серверах одновременно. Spring framework не имеет эквивалентного механизма, однако позволяет сохранять информацию между запросами, используя БД, сессию HTTP или кэш-память.

4. Выводы

В данной работе были проанализированы основные преимущества и недостатки технологий Spring Framework и Enterprise Java Bean. Выбор платформы разработки программного обеспечения хостинг-провайдера между двумя рассмотренными технологиями был сделан в пользу Spring framework в связи с тем, что эта технология позволяет вести разработку ПО более высокими темпами за счет использования лишь необходимого функционала, обеспечиваемого платформой разработки. Использование сторонних библиотек выполняется проще, что тоже сказывается на сроках разработки. Структуру кода программного средства легче изменять при использовании Spring Framewrok, что важно при прототипировании. Кроме того, программное средство может быть запущено с помощью любомого Servlet-контейнера без использования сервера приложений.

Литература

1. Мейнджер, Джейсон. JAVA: Основы программирования. [Текст]: Пер. с англ. С.Бойко под ред. Я.Шмидского / Д.Мейнджер; К.: BNV, 2003.- 570 с.

2. Перри, Брюс. JAVA сервлеты и JSP: сборник рецептов. [Текст]: Пер. с англ. / Б.Перри; М.: КУДИЦ-ПРЕСС, 2006. – 768 стр.